# TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií


Studijní program: B 2612 – Elektrotechnika a informatika

Studijní obor: 1802R022 – Informatika a logistika

# Systém bezdrátových informačních kiosků

# System of wireless information points

Bakalářská práce


Autor: Martin Patočka

Vedoucí práce: Mgr. Milan Keršláger


V Liberci dne 14.5.2011

# TECHNICKÁ UNIVERZITA V LIBERCI

**Fakulta mechatroniky, informatiky a mezioborových studií**

Akademický rok: **2010/2011**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Martin Patočka** |
| Osobní číslo: | **M07000257** |
| Studijní program: | **B2612 Elektronika a Informatika** |
| Studijní obor: | **Informatika a logistika** |
| Název tématu: | **Systém bezdrátových informačních kiosků** |
| Zadávající katedra: | **Ústav nových technologií a aplikované informatiky** |

Zásady pro vypracování:

Systém bezdrátových kiosků je distribuovaná aplikace, jejímž úkolem je obsluhovat klienty WiFi nebo BlueTooth, kteří se připojují do volně přístupné bezdrátové sítě a zajistit jejich jednoduchou správu.

Realizace:

1. Analýza problematiky
2. Vytvoření informační stránky, na kterou budou nově připojení klienti automaticky přesměrování, a která umožní jejich autentizaci v systému s ohledem na použité zařízení (notebook, PDA, mobilní telefon)
3. Vytvoření modulu správy autentizovaných klientů ve volně dostupné bezdrátové síti

| | |
|---|---|
| Rozsah grafických prací: | **dle potřeby** |
| Rozsah pracovní zprávy: | **cca 40 stran** |
| Forma zpracování bakalářské práce: | **tištěná / elektronická** |

Seznam odborné literatury:

**[W. Jason Gilmore] Velká kniha PHP a MySQL (2007)**

**[zdroje z internetu]**

| | |
|---|---|
| Vedoucí bakalářské práce: | **Mgr. Milan Keršláger**<br>Ústav nových technologií a aplikované informatiky |
| Datum zadání bakalářské práce: | **15. října 2010** |
| Termín odevzdání bakalářské práce: | **20. května 2011** |

V Liberci dne 15. října 2010

# Prohlášení

Byl(a) jsem seznámen(a) s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 o právu autorském, zejména § 60 (školní dílo).

Beru na vědomí, že TUL má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom(a) toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem TUL, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených univerzitou na vytvoření díla (až do jejich skutečné výše).

Bakalářskou práci jsem vypracoval(a) samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce.

Datum:

Podpis:

# Poděkování

Tímto bych rád poděkoval všem, kdo přispěli ke vzniku této práce. Především panu Mgr. Milanovi Keršlágerovi za jeho ochotu při vedení práce a za odborné konzultace.

# Anotace

Tato práce se zabývá možností využití distribuované bezdrátové aplikace pro potřeby prezentace firmy. Tento systém by měl být využit namísto statických informačních kiosků, tak jak je známe dnes.

Jelikož v dnešní době většina zákazníků disponuje zařízením schopným prezentace dat, jakými jsou mobilními telefony, tablety, notebooky, PDA a podobně, vznikla myšlenka, proč tyto zařízení nevyužít. Zákazník si může ve vybraných prostorech pohodlně a nerušeně prohlédnout informace, které jsou na portále umístěny. Toho je dosaženo využitím bezdrátových technologií, jakými jsou např. WiFi a BlueTooth. Díky nim zákazník nebo pracovník dokáže s informačním portálem snadno navázat spojení.

Pro vybrané uživatele portál zároveň může zpřístupnit další součásti. Například takové, které nejsou veřejné nebo které slouží pro správu systému. Uživateli je možné po autentizaci zpřístupnit i přístup na internet ve vybraných případech.

Vlastní implementace je provedena pomocí operačního systému Linux, přesněji distribucí Debian Lenny. Pro prezentaci firmy byla zvolena forma webové aplikace, která je zřejmě nejuniverzálnějším řešením.

# Anotation

This work deals with the possibility of using distributed wireless applications for presentation of the company. This system should be used instead of static information kiosks, as we know them today.

Since nowadays most customers has a device capable of presenting data, such as mobile phones, tablets, laptops, PDAs and etc. So the idea was, why not také the advantage of these devices. Customers can in selected areas conveniently and undisturbed view informations which are located on the portal. This is achieved by using wireless technologies such as WiFi and BlueTooth. With them is customer or employee easily able to establish a connection to information portal.

For selected users, the portal can also make available other parts. For example, those that are not public or are used for system administration. It is also posible to allow internet access for user in selected cases.

The actual implementation is done using the Linux operating system, specifically Debian Lenny. To promote the company was selected web application form, which is probably the most versatile solution.

# Obsah:

# Použité zkratky

| | |
|---|---|
| BNEP | BlueTooth Network Encapsulation Protocol |
| BSD | Berkeley Software Distribution |
| CSS | Cascading Style Sheets |
| DHCP | Dynamic Host Configuration Protocol |
| DNS | Domain Name System |
| DSSS | Direct Sequence Spread Spectrum |
| HTTP | HyperText Transfer Protocol |
| IEEE | Institute of Electrical and Electronics Engineers |
| IP | Internet Protocol |
| IrDA | Infrared Data asociation |
| MBWA | Mobile Broadband Wireless Access |
| MDA | Mobile Device Assistant |
| MIMO | Multiple input – Multiple output |
| NAP | Network Access Point |
| OFDM | Orthogonal Frequency Division Multiplexing |
| OS | Operating System |
| PAN | Personal Area Networking Profile |
| PDA | Personal Digital Assistant |
| PIN | Personal Identification Number |
| SQL | Structured Query Language |
| SSID | Service Set Identifier |
| SSL | Secure Sockets Layer |
| TLS | Transport Layer Security |
| VPN | Virtual Private Network |
| WAP | Wireless Application Protocol |
| WiFi | Wireless Fidelity |
| WiMAX | Worldwide Interoperability for Microwave Access |
| WPA | Wi-Fi Protected Access |
| WWW | World Wide Web |
| XHTML | eXtensible HyperText Markup Language |
| XHTML MP | XHTML Mobile Profile |

# 1. Úvod

Informační kiosky v dnešní době nacházíme na nejrůznějších místech, ať už to jsou prodejny, banky, informační centra, či různá jiná zařízení. Obvykle jsou tvořeny například jako systém s dotykovým panelem, kde si uživatel může prohlížet data, která chceme prezentovat, či v některých případech se může jednat i o běžné počítače se softwarem upraveným tak, aby uživatel měl jen tento omezený přístup.

Úkolem bakalářské práce je realizace informačního kiosku formou distribuované aplikace s bezdrátovou komunikací (BlueTooth, WiFi, WiMAX,…), kde klienty tvoří mobilní zařízení koncového uživatele (mobilní telefon, tablet, PDA, případně notebook).

Uživatel by měl být schopen se snadnou formou připojit k místnímu systému a získat tak přístup k požadovaným informacím. Je tedy třeba zajistit, aby konfigurace na straně klientského zařízení proběhla pokud možno zcela automatizovaně, tak aby byl schopný se připojit kdokoli, kdo bude mít o informace zájem. Je nutné dbát také na to, že se různá zařízení se od sebe velice liší a často některé vyžadují individuální řešení.

V neposlední řadě nesmíme zapomenout i na zabezpečení přenášených dat a to hlavně ve směru od klienta na server, kde by mohly být přenášeny některé choulostivé informace klientů.

# 2. Analýza problematiky - rozdělení na dílčí části

Pro realizaci úkolu je třeba se zaměřit na jednotlivé aspekty a najít vhodné řešení. Jelikož v dnešní době existuje velká škála OpenSource softwaru, umožňujícího značné snížení nákladů při pořizování technologie, zkusíme jej pro realizaci využít. Jeho použití je vhodné také z důvodu možnosti si upravit aplikace do jisté míry dle svých potřeb úpravou dostupných zdrojových kódů. Pokud se jedná o samotnou realizaci, je vhodné si ji rozdělit na několik vzájemně provázaných dílčích částí.

První část by se měla zabývat samotným zobrazením na klientském zařízení, jeho automatickou konfigurací s ohledem na potřeby informačního kiosku a případně vytvořením nějakého klientského softwaru, pokud to bude nutné.

Další z částí bude část komunikační. Zde musíme dořešit, jak dojde k automatickému propojení klientského zařízení a serveru, tak aby mohli spolu vzájemně komunikovat a vyměňovat si mezi sebou informace.

Třetí z nich bude část serverová, která by měla zajistit uchování potřebných dat, jejich vhodné zpracování a reprezentaci dle požadavků klienta.

## 2.1. Klientská část

V případě klientské části narážíme na velký problém v tom, že na trhu je neuvěřitelné množství rozličných zařízení, lišících se jak softwarovou výbavou, operačním systémem, tak i hardwarovou stránkou.

V případě notebooků je situace ještě celkem bezproblémová, většina z nich obsahuje podporu WiFi, některé dokonce i technologii BlueTooth a v podstatě jedinými rozšířenými OS jsou Microsoft Windows, Linux, BSD, MacOS a nyní někdy i Android.

Na poli zařízení PDA, je situace obdobná. Zde mezi běžně používané OS patří Windows Mobile, WebOS (dříve PalmOS) a BlackBerry OS. Tyto zařízení ve většině případů taktéž podporují jak připojení přes WiFi, tak pomocí BlueTooth. Dnes je začínají nahrazovat zařízení typu MDA či tablet, využívající převážně Android, WebOS a Apple iOS.

Mobilní telefony typu smartphone nebo MDA se funkčně přibližují spíše PDA, než běžným telefonům. Obvykle se u nich setkáme s operačními systémy z rodiny Windows Mobile, Windows Phone, Android, WebOS, BlackBerry OS, Apple iOS a Symbian. Ve většině případů obsahují podporu BlueTooth připojení a často u nich nalezneme i podporou pro připojení pomocí WiFi.

Další skupinou jsou běžné mobilní přístroje, kde v podstatě neexistuje žádný jednotný operační systém. Často se jedná o naprosto proprietální řešení postavené na míru hardwaru telefonu. Pokud se jedná o nějaké univerzální řešení, bývá uživateli znemožněn přístup jakkoli do systému přidávat nové funkce, či ho nějak pozměnit. Jedinou volbou pro tuto skupinu mobilních telefonů je zřejmě programovací jazyk Java, který bývá u většiny novějších přístrojů podporován, jako možnost rozšíření funkcionality zařízení. U těchto telefonů to není ani jednoduché co se týče bezdrátové komunikace. BlueTooth sice bývá obsažen, ale ne vždy umožňuje napojení na úrovni síťového protokolu. A s WiFi se u starších nebo levnějších modelů setkáváme jen zřídka.

Asi jediná a pro nás zásadní vlastnost, ve které se většina těchto zařízení shoduje, je možnost přístupu na webové stránky. V podstatě na všech zařízeních nalezneme nějakou formu webového prohlížeče, či u velmi starých modelů alespoň prohlížeče WAP stránek. Toho tedy lze dobře využít jako rozhraní pro zobrazení dat, které bychom chtěli zákazníkovi poskytovat.

Vytvářet nějakou formu klientské prezentační aplikace pro různá zařízení by bylo velice neefektivní a díky různorodosti platforem a uzavřenosti některých z nich, by se jednalo o velmi nešikovné řešení. Využití integrovaného webového prohlížeče tedy bude optimálním řešením problému. Je třeba dořešit navázání spojení a jak zařízení automaticky nastavit tak, aby začalo načítat data z našeho informačního serveru. Touto problematikou se budeme zabývat následně.

## 2.2. Komunikační část - technologie, zabezpečení, automatizace navázání

Pokud se podíváme na možnosti zařízení, které očekáváme, že koncový uživatelé budou používat, nalezneme v podstatě jen několik málo použitelných technologií vhodných pro naše potřeby. IrDA se nehodí kvůli velmi krátkému dosahu a nabízí se tedy jen WiFi, BlueTooth a případně některá z přicházejících technologii, jakými jsou WiMAX (IEEE 802.16) a MBWA (IEEE 802.20).

Technologie WiMAX a MBWA zatím nejsou u nás vůbec rozšířené a i celosvětově, je velmi málo koncových zařízení, které jí podporují, proto ji prozatím nebudeme brát v potaz. Její implementace by byla v podstatě naprosto totožná s tou, kterou se budeme zabývat u WiFi.

### 2.2.1. Technologie WiFi

WiFi je zkratkou slov Wireless Fidelity (bezdrátová věrnost). Jedná se o technologii primárně navrženou pro lokální bezdrátové sítě. Je určena převážně pro domácnosti, kanceláře, haly a další vnitřní prostory. I přesto ji některé firmy využívají i ve venkovním prostředí, díky tomu, že výrobci s postupem času umožnili nastavení časování i tak, že zvládá delší dosahy. Existuje několik standardů, se kterými se můžeme setkat.

Původní IEEE 802.11 využívalo DSSS modulaci (Direct Sequence Spread Spectrum - technika přímého rozprostřeného spektra) a umožňovalo teoretickou přenosovou rychlost okolo 2Mbps. Specifikace vznikla v roce 1999. Tato norma pracovala v pásmu 2,4 GHz.

Následovala norma 802.11b ve stejném roce, která přinesla navýšení propustnosti až na 11Mbps.

Později se v roce 2001 objevila norma 802.11a určena pro pásmo 5 GHz, využívající již OFDM (*Orthogonal Frequency Division Multiplexing*, česky *ortogonální multiplex*

s *kmitočtovým dělením*.). Jedná se o přenosovou techniku pracující s tzv. rozprostřeným spektrem, kdy je signál vysílán na více nezávislých frekvencích, což zvyšuje odolnost vůči interferenci. [6] Díky tomu dosahuje teoretické rychlosti 54Mbps. Bohužel v době vydání normy nebyli ještě firmy schopny dodat hardware za rozumných finančních podmínek, a proto musela vzniknout norma méně náročná.

Tou byla v roce 2003 norma 802.11g pracující na nižší frekvenci 2,4 GHz a umožňující komunikaci teoretickou rychlostí okolo 54Mbps díky podpoře OFDM modulace. Tato norma zachovává zpětnou kompatibilitu s 802.11b a podporuje tedy i DSSS modulaci (pokud protistrana neumožňuje rychlejší OFDM).

Nejnovějším používaným standardem je norma 802.11n. V tuto chvíli se na trhu běžně prodávají zařízení vyrobené dle specifikací 802.11n Draft 2.0, s teoretickou přenosovou rychlostí okolo 300Mbps. Reálná rychlost je okolo 200Mbps. Teoreticky se do budoucna počítá až se zdvojnásobením propustnosti. Větších přenosových rychlostí je dosaženo použitím technologie MIMO, kdy se jak na straně přijímače, tak na straně vysílače využívá více anténních jednotek. Zároveň došlo k úpravě fyzické a linkové vrstvy, tak aby se dosahovalo vyšší efektivity při přenosu. Počítá se s využitím pásma 2,4 GHz i 5 GHz. Zařízení, která dnes podporují normu 802.11n jsou zpětně kompatibilní i s normami 802.11a/b/g.

Pro připojení koncových zákazníků by sloužil tzv. access point neboli přístupový bod. Ten zajišťuje samotné připojení zákazníků a předání informací po standardním ethernetovém rozhraní. Zároveň je schopný zajistit jistou míru zabezpečení jakými jsou například WPA/WPA2 (Wi-Fi Protected Access). I když v tomto případě, by měl access point zůstat bez zabezpečení, aby se mohl připojit kdokoli i bez znalosti přístupového hesla. Jelikož zařízení mají často podporu jen pro pásmo 2,4 GHz (nemusí obsahovat oba typy antén), bude lepší využít tuto frekvenci.

Jako vhodnou hardwarovou platformu bych volil například produkty RouterBoard lotyšské firmy MikroTik. V tuto chvíli pro ně existují bezdrátové karty R52 stejnojmenného výrobce umožňující vysílání v normách 802.11a/b/g i nové karty R52N podporující navíc normu 802.11n s využitím dvou vysílacích antén. Tato platforma se osvědčila po celém světě a například u nás v České Republice je jednou z nejrozšířenějších na poli menších poskytovatelů

bezdrátového připojení k internetu. Zařízení umožňují podrobnou diagnostiku okolního vysílání. Je možné je nechat například připojit k serveru zabezpečenou VPN a díky tomu nemusí být server dostupný přímo v dané lokalitě. Dále umožňují nastavit DHCP server pro automatické přidělování IP adresy a dalších parametrů zákazníkům. Dokáže zařídit i odchytávání paketů pomocí nastavení „proxy-arp“. To poté umožní teoreticky poskytnout navázání spojení i v případě, kdy zákazník má staticky chybně nakonfigurovanou síťovou kartu. Díky tomu, že se paket zachytí, jsme schopni zjistit, o jaký požadavek se jednalo a vhodnou formou ho zpracovat.

Další variantou je instalace bezdrátové karty přímo do serveru a řešení access pointu softwarově. Toto řešení přináší o trochu složitější konfiguraci a omezuje nás v možnosti rozmístění antén díky tomu, že koaxiální kabely by neměly být delší než cca 10 m, jinak dochází k velkým ztrátám na signálu. Naproti tomu řešení v podobě hardwarového access pointu nás v tomto téměř nelimituje, jsme omezeni pouze specifikacemi ethernetu.

### 2.2.2. Technologie BlueTooth

Pro připojení pomocí BlueTooth je asi jedinou vhodnou volbou využití profilu PAN (Personal Area Networking Profile), který se snaží emulovat běžnou síťovou kartu a je tedy možné po něm přenášet data jako po běžném ethernetu. Na Linuxu jeho funkčnost zajišťuje skupina softwaru z projektu BlueZ. Jedná se o oficiální BlueTooth podporu pro platformu Linux. Pro zajištění funkčnosti tak jak ji potřebujeme, je nutné, aby byl do jádra systému zaveden modul "bnep" (BlueTooth Network Encapsulation Protocol).

Například na distribucích založených na Debianu bychom měli nainstalovat komponenty bluez-utils, bluez-compat a bluetooth pomocí balíčkovacího systému. Pro autokonfiguraci zařízení bude třeba ještě balíček DHCP serveru, např. dhcp3-server. Tuto instalaci snadno provedeme například pomocí integrovaného nástroje apt-get, popřípadě grafického Synaptic. Balíček bluetooth obsahuje základní podporu pro zařízení typu BlueTooth. Bluez-utils přináší základní nástroje pro navázání spojení, vyhledávání okolních zařízení a podobně. Bluez-compat zvyšuje kompatibilitu přidáním dalších profilů, tento balík obsahuje i službu PAN. Dále se jedná již jen o správnou konfiguraci, kterou se budeme zabývat později.

## 2.3. Serverová část - volba platformy, HTTP server, reprezentace dat a jejich uložení

Serverová část se skládá z operačního systému, z databázového systému pro uchování dat, skriptů pro jejich zpracování a serveru pro následnou komunikaci s klientskými zařízeními. Nyní si popíšeme jednotlivé části.

### 2.3.1. Operační systém

Pro část serveru připadají v úvahu řešení na základě Microsoft Windows Server, linuxové distribuce nebo nělterý blíže "UNIXově" založený OS jakým je např. OpenBSD, FreeBSD či Solarix. Jelikož budeme po serveru požadovat i některé služby spojené s navázáním spojení ne úplně běžnou cestou, bude dle mých zjištění nejsnazší využít některé otevřené distribuce, která umožňuje značné přizpůsobení chování hardwaru jakými je např. BlueTooth či bezdrátová síťová karta.

Při zkoumání možností narazíme na velké množství nejrůznějších distribucí. Nabízí se například distribuce BSD, ať už se jedná o rozšířené FreeBSD nebo jeho alternativu OpenBSD. Zde je jen menší problém v tom, že distribuce jsou hodně zaměřená na práci s tzv. "porty". Jedná se o systém připravených kompilačních skriptů, kdy se v podstatě všechny aplikace při aktualizacích a instalacích překládají ze zdrojových kódů. Existují sice softwary, které toto automatizují a umožňují tak téměř bezobslužnou funkci. I přesto se však jedná o zbytečně velkou zátěž stroje a případné komplikace při nějaké neshodě v závislosti verzí. Toto řešení by také vyžadovalo řádně kvalifikovanou údržbu.

Jako další vhodné řešení se jeví některý z operačních systémů rodiny Linux. Velkou výhodou je ve většině případů velmi vydařený balíčkovací systém, kdy jsou v podstatě všechny aktualizace promítnuty do závisle potřebných balíčků uložených na repositářovém serveru. Tedy pokud udržíme všechny balíky v aktuálních verzích, mělo by to zajistit ochranu proti bezpečnostním dírám a zároveň bezkonfliktnost mezi jednotlivými aplikacemi a částmi systému.

Jako vhodné kandidáty bych volil buď distribuci Debian, případně Ubuntu Server pro velmi uživatelsky přívětivou formu prostředí a kvalitní balíčkovací systém nebo například Fedora Core či CentOS, jakožto klon Red Hat Enterprise Linux s velmi dobrou podporou a vysokou stabilitou.

Linux jako takový má obecně velmi dlouhou tradici obzvláště na serverových strojích, kde se osvědčil pro jeho stabilitu a bezpečnost. Využívá ho dnes většina hostingových firem společně s Windows Server a nalezneme ho i v některých intranetových systémech. Nebál bych se tedy jeho nasazení.

### 2.3.2. WWW/WAP - HTTP server, skriptovací programovací jazyk

Jako velmi rozšířený a vhodný webový server, díky mnoha nejrůznějším modulům a širokým možnostem konfigurace se jeví Apache HTTP Server Project. Je možné jej využít jak pro poskytování obsahu pro běžné webové prohlížeče tak formou WAP stránek pro starší mobilní telefony, které neumožňují zobrazení plnohodnotných webových stránek. Záleží tedy čistě na webmasterovi, jakou formou bude obsah poskytovat.

Apache umožňuje též využití TLS/SSL šifrování mezi serverem a klientem, pokud to daný klient umožňuje. Díky jeho častému využití u webhostingových firem, není problém pro tento HTTP server nechat vystavit i kompatibilní ověřené certifikáty od renomovaných firem jakými jsou Thawte, Verisign či další na trhu. Tento typ zabezpečení zaručuje autentičnost serveru. Díky jednoznačnému podpisu zaručíme klientovi, že komunikuje skutečně s tím serverem, se kterým chce. Zároveň nám tyto metody umožní šifrovaný přenos dat na vysoké úrovni. Je možné využít asymetrickou šifru, kdy každá strana má svůj unikátní šifrovací a dešifrovací klíč. Kvůli zpětné kompatibilitě je samozřejmě možné využít i jen symetrické šifrování, či jednosměrné hashovaní. Ale bezpečnost přenášených dat je poté o řád nižší. Pro navázání spojení a nalezení nejvyššího stupně společně podporované ochrany se využívá tzv. TLS/SSL Handshake.

Typická inicializace TLS probíhá následovně: [11]

- Klient pošle zprávu **ClientHello** oznamující nejvyšší verzi TLS, kterou podporuje, náhodné číslo a seznam doporučených šifrovacích sad a kompresních metod. [11]

- Server odpoví zprávou **ServerHello** obsahující zvolenou verzi protokolu, náhodné číslo, šifrovací a kompresní metodu vybranou z klientem nabídnutého seznamu. [11]

- Server pošle svůj certifikát (**Certificate**), pokud to zvolená šifra umožňuje. Současné certifikáty jsou založeny na X.509, ale existuje návrh na používání certifikátů vycházejících z OpenPGP. [11]

- Server může pomocí *CertificateRequest* vyžadovat certifikát od klienta, aby bylo spojení autentizováno vzájemně. [11]

- Server pošle zprávu **ServerHelloDone**, která signalizuje, že ukončil iniciační dohodu na používaných mechanismech. [11]

- Klient odpoví zprávou **ClientKeyExchange**, jež může obsahovat *PreMasterSecret,* veřejný klíč nebo nic (v závislosti na zvolené šifře). [11]

- Klient a server následně z náhodných čísel a PreMasterSecret pomocí pečlivě navržené pseudonáhodné funkce vypočítají „master secret“. Veškeré ostatní klíče jsou odvozeny z něj (a z generovaných náhodných hodnot). [11]

- Klient nyní odešle zprávu **ChangeCipherSpec**, jíž v podstatě sděluje „veškerá další data od mne budou šifrována“. Za pozornost stojí, že ChangeCipherSpec je sám o sobě protokolem záznamové vrstvy s typem 20, nikoli 22. [11]

- Na závěr klient pošle šifrovanou zprávu **Finished** obsahující hash a MAC předchozích iniciačních zpráv. [11]

- Server se pokusí dešifrovat klientovu zprávu Finished a ověřit její hash a MAC. Pokud dešifrování či ověření selže, inicializace je považována za neúspěšnou a spojení by mělo

být ukončeno. [11]

- Konečně server pošle zprávy **ChangeCipherSpec** a svou zašifrovanou **Finished** a klient provede analogické dešifrování a ověření. [11]

- V tomto okamžiku je inicializace dokončena a je povolen aplikační protokol, jehož typem obsahu je 23. Aplikační zprávy vyměňované mezi klientem a serverem budou zašifrovány. [11]

V dnešní době moderních skriptovacích jazyků jakými jsou PHP, Perl, Python, ASP.NET a mnohé další je již celkem běžnou věcí, že se stránka automaticky přizpůsobí podle toho, z jakého typu prohlížeče přichází žádost o informace. Většina prohlížečů při žádosti o určitou stránku doplňuje do hlavičky i informace o tom, o jaký prohlížeč se jedná, či jaké kódování podporuje. Díky tomu je tedy možné jedním kódem generovat stránku jak pro běžné WWW prohlížeče, tak mírně upravenou variantu pro prohlížeče typu WAP.

Jako vhodné skriptovací programovací jazyky pro naše účely připadají v úvahu v podstatě všechny jmenované kromě ASP.NET, který je úzce spjatý s platformou ISS od Microsoftu a na použití ve spolupráci s HTTP Serverem Apache se příliš nehodí. Pro PHP, Perl a Python existují přímo do Apache oficiální moduly a jejich integrace tedy není nijak problematická. Ve všech třech případech se jedná o moderní programovací jazyky, které jsou objektově orientované, umožňují využití dynamicky alokovaných proměnných, snadný přístup k nejrůznějším databázovým systémům a obsahují velké množství již vytvořených funkcí pro nejrůznější operace. Výběr tedy záleží hlavně na požadavcích webmastera a na implementaci z hlediska napojení (například na stávající webové stránky).

### 2.3.3. Databázový systém

V dnešní době se do popředí čím dál tím častěji dostávají databázové systémy. Jejich hlavní výhodou je komplexnost a možnost práce několika odlišných programů nad jedněmi daty současně. Zároveň nám některé dnešní databázové systémy umožňují i takové věci jakými jsou automatické synchronizace vzdálených systémů, replikace pro případ výpadku nebo režimu master-slave. Režim master-slave je takový, kdy několik systémů se vzájemně synchronizuje a umožňuje jak čtení, tak změny a tím pádem možnost přípravy obsahu. A několik dalších systému na ně napojených si jen přebírá obsah a dále jej zpřístupňuje.

Mezi dnes nejpoužívanější OpenSource databázové systémy se řadí především MySQL a PostgreSQL. Z těch komerčních pak jistě stojí za zmínku Oracle nebo Microsoft SQL.

PostgreSQL se vyznačuje větší komplexností a rychlejším zpracováním v případě velkých objemů dat, nebo při složitých operacích. Umožňuje také vytváření interních funkcí na vysoké úrovni a řadu dalších pokročilých funkcí. Celá databáze je vedena pod BSD licencí a je tedy naprosto volně použitelná. Jedinou nevýhodou jsou značné problémy s vytvářením clusterů, replikací a dalších provázání. Existuje několik externích řešení, ale vždy hrozí při větší zátěži (větším objemu dat), že dojde k selhání replikace. Při použití jako databáze na jednom serveru, ji však lze jen doporučit. Pro provoz na dvou strojích v režimu master-master, či master-slave (i více slave nodů) lze po určitých úpravách využít projekt Bucardo, který pracuje v nové verzi celkem stabilně, dle mých zkušeností.

MySQL je na rozdíl od PostgreSQL původně jednoduchá a rychlá databáze, i když to v poslední době již není příliš pravda a většina složitějších funkcí byla doimplementována dodatečně. Hodí se hlavně na jednoduché operace, kde podává vynikající výkony. Pro složitější úkoly napříč velkým množstvím tabulek, či pro složité napojování a podobně se příliš nehodí. Velkou výhodou je zde nativní podpora pro master-slave režim, kdy je možné jej nastavit i zároveň v obou směrech zároveň a tím získat režim master-master. Hodí se tedy jako bezplatná platforma pro budování databázových clusterů jednoduchých databází, i když se zde občas shledáváme s problémy.

Oracle je velmi dobrou databází s mnohaletými zkušenostmi. Má bezproblémovou podporu pro Linux a je vhodná zejména pro produkční nasazení, díky možnostem jejího rozšíření do clusteru, decentralizaci a technické podpoře. Nevýhodou je snad jen nutnost pořízení licence. Jedná se totiž o komerční produkt.

### 2.3.4. IPtables

IPtables je softwarový firewall běžně dostupný v linuxových distribucích. Společně s povoleným směrováním paketů ho lze využít i pro složitější úkoly při směrování paketů, jakými jsou funkce NAT, či přesměrování. V našem případě by bylo vhodné jej použít pro omezení provozu klientů jen pro účely prohlížení obsahu, který jim chceme poskytnout. Popřípadě jej můžeme využít pro přesměrování na náš obsah při pokusu o přístup někam jinam.

### 2.3.5. Webová aplikace

Měla by být napsána v jednom z navrhovaných jazyků, ať už se jedná o PHP, Perl či Python.

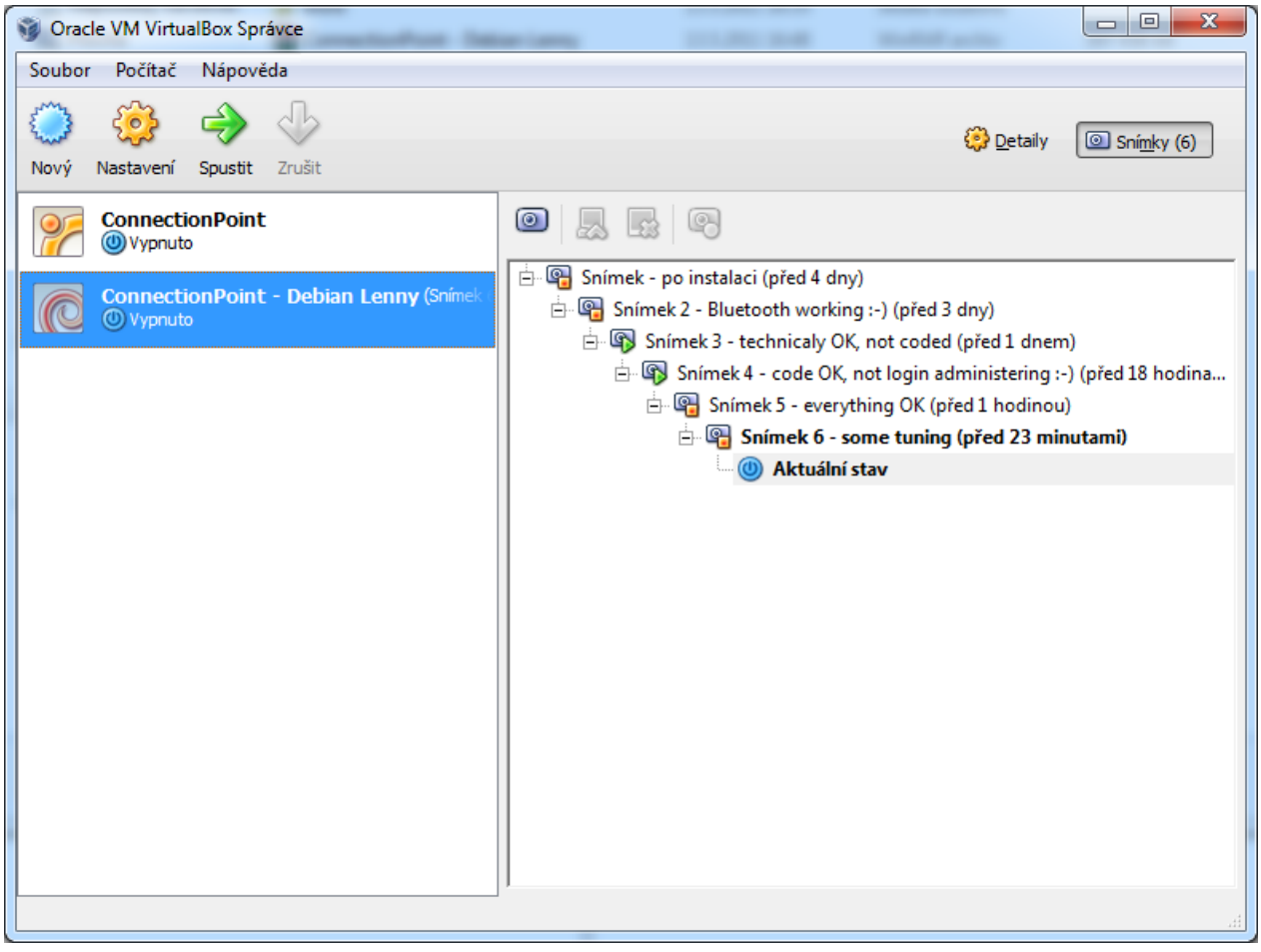
Dle typu zařízení uživatele musí webová aplikace vhodně zvolit výstupní formát pro předání dat. Ve většině případů bude pro formátování webových stránek nejvhodnější použít jazyk XHTML, doplněný o CSS styly dotvářející vzhled stránek. Takto navržené stránky by měly být zobrazitelné na absolutní většině dnes dostupných mobilních zařízení. Ať už se jedná o notebooky, tablety, PDA, či mobilní telefony. Pro zařízení bez podpory prohlížet běžné HTML stránky a mající pouze podporu WAP, bych doporučil použití jazyku XHTML MP. Dnes už ale taková zařízení na trhu vůbec nejsou.

# 3. Realizace systému

Pro realizaci systému jsem se nakonec rozhodl využít linuxovou distribuci Debian Lenny, kvůli dobrému portfoliu podporovaných aplikací a kvalitnímu balíčkovacímu systému. Další výhodou je, že systém využívá bluez-utils ještě ve verzi 3.x, které jsou lépe zdokumentované a budou pro naši implementaci jednodušší.

Pro tvorbu webové aplikace jsem využil jazyka PHP z důvodu jeho znalosti a dobrých zkušeností s ním. Databázový systém jsem zvolil PostgreSQL. Princip fungování a ověřování uživatelů bude popsán v následující kapitole.

Celý systém jsem odladil pod virtualizačním nástrojem VirtualBox společnosti Oracle Corporation (Obrázek č.1).




**Obrázek č.1**

Finální obraz systému je přiložen na disku CD. Přístupové jméno správce je „root" a heslo „connectionpoint". Lze jej tedy bez složitého instalování a nastavování začít hned využívat.

## 3.1. Webová aplikace

Aplikace byla vytvořena v jazyce PHP a využívá možnosti spouštět konzolové příkazy za pomocí funkce „shell_exec". Je přiložena na CD v archivu *www.rar.*

Uživatel je po připojení automaticky nasměrován na tuto aplikaci. Ta se poté postará o zobrazení příslušného obsahu, dle typu zařízení klienta a případnou úpravu pravidel firewallu.

Za pomoci webové aplikace je možná i jednoduchá správa uživatelů včetně nastavení příslušných oprávnění.

Princip její funkce je takový, že ve firewallu jsou přednastavena pravidla neumožňující komunikovat uživatelům mimo server (REJECT spojení ve FORWARD tabulce FILTERu IPtables) a zároveň pravidlo, které veškerá TCP spojení přesměrovává na interní webový server, kde je aplikace obsluhuje (využití DNAT v PREROUTINGu IPtables).
*(nastavení je popsáno v rámci sekce 3.2.7. Implementace webové aplikace)*

Webové stránky prezentující firmu by měly být umístěny do složek „/var/www/web-mobile/" (verze pro mobilní zařízení) a „/var/www/web-desktop" (verze pro osobní počítače). Aplikace pak na základě rozhodnutí o jaký typ zařízení se jedná, nasměruje uživatele na správnou prezentaci.

Pokud má být některému uživateli umožněn přístup na internet, dojde k vložení pravidel povolujících průchod paketů a také je nastavena vyjímka pro přesměrování. Uživatel po tomto povolení může na portál přistupovat za pomocí DNS jména, které je přednastaveno na „connection-point.web".

Autentifikace uživatele je zajištěna oproti PostgreSQL databázi a v rámci kódu je realizována třídou Session (*/var/www/classes/access.class.php*). Tato třída využívá PHP funkcí pro vytváření a správu „Sessions“. Tedy pro trvalé uchování obsahu proměnných na základě identifikace uživatele. To umožňuje trvalé ověření po dobu jeho připojení. Dočasné soubory, které jsou k jejich použití zapotřebí, jsou ukládány do složky */var/www-sessions/*.

Pro zjištění typu klientského prohlížeče a jeho schopností se využívá interní funkce PHP „get_browser“, která je doplněna o příslušný aktuální konfigurační soubor.
*(nastavení je popsáno v rámci sekce 3.2.5. Konfigurace webového serveru)*

Aby bylo možné zasahovat do tabulky „iptables“ (firewall) a využívat příkazu „arp“ (tabulka ARP) je nutno využít nástroje „sudo“. Ten umožňuje vybraným uživatelům spouštět určené příkazy s právy administrátora. V našem případě se jedná o příslušná oprávnění pro uživatele „www-data“. *(nastavení je popsáno v rámci sekce 3.2.7. Implementace webové aplikace)*

Aby nedocházelo k zahlcování IPtables již nepotřebnými pravidly, byl navržen mechanismus, díky kterému je možné je pravidelně čistit. Pokud se provede čištění (znovunačtení defaultní konfigurace), způsobí to obnovení nasměrování všech spojení na webovou aplikaci. Ta rozpozná, na jakou stránku chtěl uživatel přistupovat, ověří, zda má oprávnění přistupovat na internet, obnoví pravidla a vyvolá její znovunačtení.

Pro zvýšení bezpečnosti jsou pravidla do IPtables zadávány včetně MAC adresy. Tím je zajištěno, že pokud by někdo použil stejnou IP (například zadanou staticky), firewall by mu přístup neumožnil.

Funkce pro nastavení IPtables jsou součástí třídy IpTables a umožňují přidávání a odebírání příslušných pravidel a ověření jejich zadání.

*Soubor **/var/www/classes/iptables.class.php:***

```
class IpTables {
        function allowInternetAccess($ip, $mac) {          //Přidání pravidel
            shell_exec("sudo iptables -t nat -I PREROUTING 1 -s $ip/32 -m mac --mac-source $mac -j ACCEPT");
            shell_exec("sudo iptables -t filter -I FORWARD 1 -d $ip/32 -i eth0 -j ACCEPT");
            shell_exec("sudo iptables -t filter -I FORWARD 1 -s $ip/32 -m mac --mac-source $mac -o eth0 -j ACCEPT");
            return;
        }
        function denyInternetAccess($ip, $mac) {          //Odstranění pravidel
            shell_exec("sudo iptables -t nat -D PREROUTING -s $ip/32 -m mac --mac-source $mac -j ACCEPT");
            shell_exec("sudo iptables -t filter -D FORWARD -d $ip/32 -i eth0 -j ACCEPT");
            shell_exec("sudo iptables -t filter -D FORWARD -s $ip/32 -m mac --mac-source $mac -o eth0 -j ACCEPT");
            return;
        }
        function isInternetAccess($ip) {    //Ověření zda je IP adresa již zadaná
            if ($iptables_info = explode("\n", shell_exec("sudo iptables -t nat -S"))) {
                foreach ($iptables_info as $rule) {
                    if ($found = strpos($rule, $ip)) {
                        $rule_info = preg_split("/[\s,]+/", $rule);
                        return $rule_info[7];      //return MAC
                    }
                }
            }
            return false;
        }
        function getRulesList() {              //Funkce vrací pole s nastavenými pravidly
            $return['mangle'] = shell_exec("sudo iptables -t mangle -L");
            $return['nat'] = shell_exec("sudo iptables -t nat -L");
            $return['filter'] = shell_exec("sudo iptables -t filter -L");
            return $return;
        }
    }
```

## 3.2. Instalace a konfigurace systému

Pro instalaci a provoz systému byl zvolen vizualizační nástroj VirtualBox. To umožňuje snadný přenos systému na jiný hardware a zároveň provoz společně s jinými službami (popřípadě operačnímy systémy) na jednom stroji.

VirtualBox v aktuální verzi 4.x již umožňuje mapování USB zařízení do virtuálního stroje, což jsme využili pro připojení adaptéru BlueTooth do našeho systému.

Instalace a konfigurace systému probíhala v několika následujících krocích.

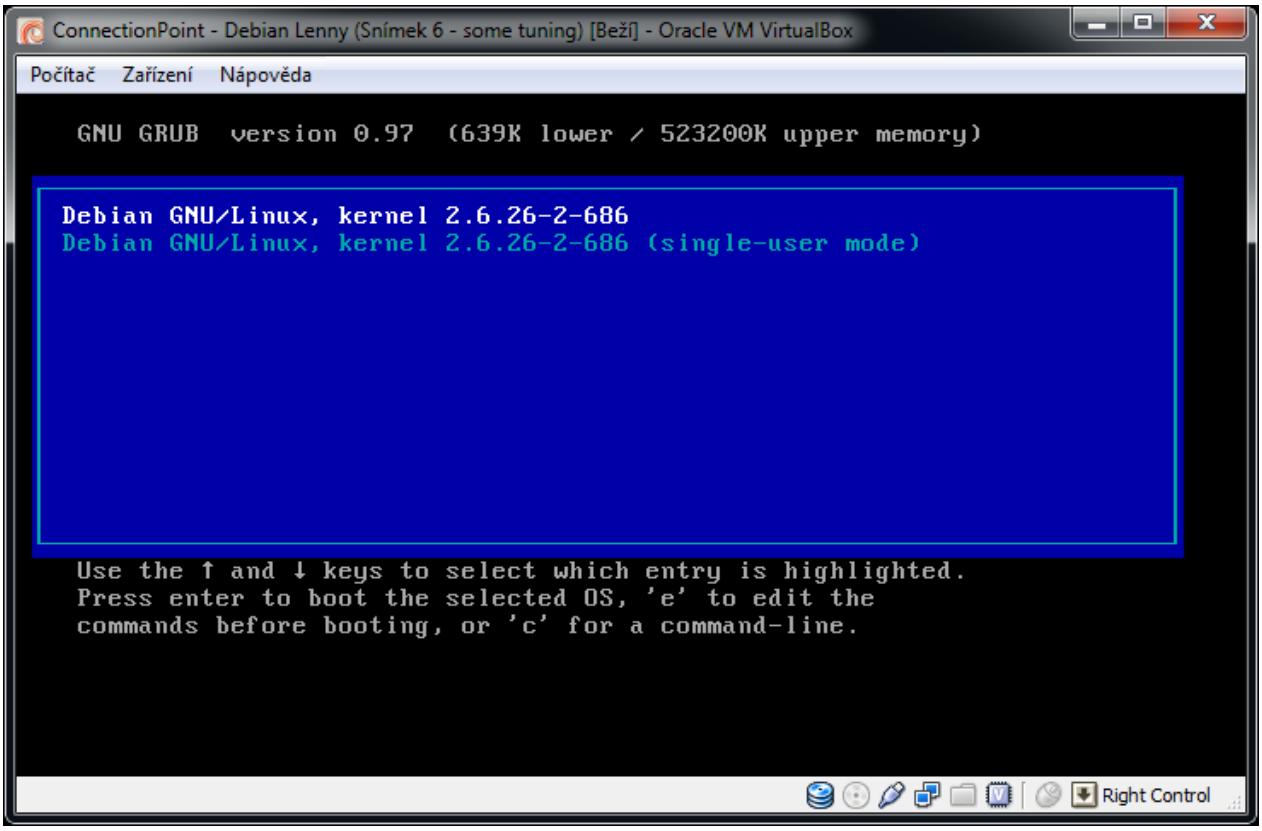
### 3.2.1. Instalace operačního systému

Ve VirtualBoxu si vytvoříme nový virtuální stroj, kde zvolíme typ Linux a distribuci Debian. Stroji přidělíme dvě síťové karty, jednu pro připojení na internet a druhou pro vnitřní síť umožňující připojení klientů. Velikost disku můžeme nechat standardní, tedy 8GB.

Pokud chceme využívat připojení přes BlueTooth, je nutné ještě namapovat příslušný USB BlueTooth adaptér do virtuálního stroje.

Poté provedeme běžnou instalaci distribuce Debian Lenny z instalačního CD nebo ISO obrazu. Stačí nám instalace v textovém režimu, se základní sadou balíčků. Celou instalací nás provede přehledný průvodce.

Po instalaci by nám již měl nastartovat systém ve vizualizovaném prostředí (Obrázek č.2).




**Obrázek č.2**

## 3.2.2. Instalace potřebných balíčků

Po instalaci je nutné doplnit některé další součásti systému a aplikace. To provedeme pomocí nástroje „apt-get“, který je standardní součástí systému.

Instalace probíhá pomocí příkazu „apt-get install *jmeno-baliku*“, kde můžeme uvést i více balíků najednou, oddělených mezerou.

Je nutno, aby byli doinstalovány tyto součásti:

Webový server Apache 2.2:

| | |
|---|---|
| apache2 | install |
| apache2-utils | install |
| apache2.2-common | install |

PostgreSQL Databáze:

| | |
|---|---|
| postgresql | install |
| postgresql-client | install |

Podpora PHP pod Apache a modul pro PostgreSQL databázi:

| | |
|---|---|
| libapache2-mod-php5 | install |
| php5-common | install |
| php5-pgsql | install |

Utility a součásti pro konfiguraci BlueToothu:

| | |
|---|---|
| bluez-hcidump | install |
| bluez-utils | install |
| libbluetooth2 | install |

Jednoduchý DHCP server a DNS relay démon:

| | |
|---|---|
| dnsmasq | install |

Aplikace pro správu stavové tabulky otevřených spojení:

| | |
|---|---|
| conntrack | install |

Po nainstalování těchto balíčků přejdeme na jejich konfiguraci v dalších krocích.

### 3.2.3. Konfigurace BlueTooth

Pro správnou funkci BlueTooth adaptéru je možné využít funkce mapování USB zařízení do virtuálního stroje (platí jen pro provoz pod VirtualBoxem).

Abychom se mohli na BlueTooth připojit pomocí profilu NAP a využívat tak síťového protokolu, je třeba několik konfiguračních změn, které nyní uvedeme. Důležité řádky budou uvedeny tučně.

*Soubor **/etc/defaults/bluetooth***

***BLUETOOTH_ENABLED=1***
*HID2HCI_ENABLED=1*
*HIDD_ENABLED=0*
*HIDD_OPTIONS="--master --server"*
*DUND_ENABLED=0*
*DUND_OPTIONS="--listen --persist"*
***PAND_ENABLED=1***
***PAND_OPTIONS="--listen --role NAP --devup /etc/bluetooth/pan/dev-up"***
***SDPTOOL_OPTIONS="add NAP"***

*Soubor **/etc/bluetooth/hcid.conf***

```
#
# HCI daemon configuration file.
#

# HCId options
options {
        # Automatically initialize new devices
        autoinit yes;

        # Security Manager mode
        #   none - Security manager disabled
        #   auto - Use local PIN for incoming connections
        #   user - Always ask user for a PIN
        #
        security auto;

        # Pairing mode
        #   none  - Pairing disabled
        #   multi - Allow pairing with already paired devices
        #   once  - Pair once and deny successive attempts
        pairing multi;

        # Default PIN code for incoming connections
        passkey "1234";
}

# Default settings for HCI devices
device {
        # Local device name
        #   %d - device id
        #   %h - host name
        name "%h-%d";
```

```
        # Local device class
#       class 0x000100;
        class 0x020100;

        # Default packet type
        #pkt_type DH1,DM1,HV1;

        # Inquiry and Page scan
        iscan enable; pscan enable;

        # Default link mode
        #   none    - no specific policy
        #   accept - always accept incoming connections
        #   master - become master on incoming connections,
        #            deny role switch on outgoing connections
        lm accept, master;

        # Default link policy
        #   none     - no specific policy
        #   rswitch - allow role switch
        #   hold     - allow hold mode
        #   sniff    - allow sniff mode
        #   park     - allow park mode
        lp rswitch,hold,sniff,park;
}
```

Soubor **/etc/bluetooth/network.conf**

```
Interface=bnep0
```

Soubor **/etc/bluetooth/pan/dev-up**

```
#!/bin/sh
ifup --force $1
```

Takto by měla být konfigurace BlueTooth adaptéru kompletní. Stačí jen dokončit nastavení síťových rozhraní a DHCP serveru a bude možné navázat připojení.

### 3.2.4. Konfigurace síťových rozraní, DHCP serveru a DNS

Síťová rozhraní se konfigurují v Linuxu pomocí textového souboru, jeho obsah může být například následující. IP adresy je samozřejmě možné přizpůsobit, jen je nutné myslet na to, aby se upravil i příslušný konfigurační skript pro IP tables.

*Soubor **/etc/network/interfaces***

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo eth1
iface lo inet loopback

# The primary network interface
allow-hotplug eth0
iface eth0 inet dhcp

# Bluetooth Connections
iface bnep0 inet static
   address 192.168.66.1
   netmask 255.255.255.0

#WiFi Connections
iface eth1 inet static
   address 192.168.77.1
   netmask 255.255.255.0
```

Dále je třeba na příslušných rozhraních spustit DHCP server, aby připojeným zařízením mohl přidělovat IP adresy. Pro tento účel využijeme softwaru DNSmasq, který zároveň slouží jako jednoduchý DNS relay agent a má výhodu v tom, že je možné ho spustit i pokud není interface v danou chvíli k dispozici. V konfiguračním souboru nám postačí nastavení základního parametru, a to na jakém rozsahu má přidělovat IP adresy.

*Soubor* ***/etc/dnsmasq.conf***

***dhcp-range=192.168.66.50,192.168.66.150,12h***
***dhcp-range=192.168.77.50,192.168.77.150,12h***

Tím by mělo být po stránce síťové vrstvy vše připraveno. Konfigurace předpokládá vstupní interface „eth0“, s tím že adresa na něj bude přidělena pomocí DHCP. Pokud tomu bude jinak, je nutno konfiguraci upravit.

## 3.2.5. Konfigurace webového serveru

Webový server Apache 2.2 nám bude vyhovovat víceméně v základní konfiguraci, pokud nebudeme vyžadovat SSL připojení. Pokud ano, je nutné ještě navíc oproti našemu příkladu nakonfigurovat certifikáty.

Z důvodu toho, že na server mohou být přesměrována různá URL, na které chce klient přistupovat, je nutné tyto požadavky odchytit. Nejsnazším řešením je konfigurace souboru 404, který udává na jaký soubor klienta nasměrovat, pokud jeho žádaná stránka nebyla nalezena. To provedeme jednoduše následovně.

*Soubor* ***/etc/apache2/conf.d/errors***

***ErrorDocument 404 /index.php***

Dále je dobré, aby server nezobrazoval výpis souborů a adresářů, pokud není požadován konkrétní soubor (tzv. „Directory Listing“). To jednoduše provedeme pozměněním konfigurace adresáře (musíme odstranit Options – Indexes).

*Nastavit v* ***/etc/apache2/sites-enabled/000-default***

```
<Directory /var/www/>
        Options FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
</Directory>
```

Dále je nutné provézt několik malých změn v konfiguraci PHP, abychom zpřístupnili funkci „get_browser“, vypnuli „safe_mode“ a povolili použití globálních proměnných pro použití se „Sessions“.

*Nastavit v* ***/etc/php5/apache2/php.ini***

```
safe_mode = Off

[browscap]
browscap = /etc/php5/extra/browscap.ini

[Session]
session.bug_compat_42 = 1
session.bug_compat_warn = 0
```

Funkce „get_browser“ slouží pro rozpoznání parametrů prohlížeče a vyžaduje k tomu aktuální konfigurační soubor, který je platný pro PHP. Ten můžeme získat například na této adrese: *http://browsers.garykeith.com/downloads.asp*
Tento soubor je nutné umístit takto: ***/etc/php5/extra/browscap.ini*** *(viz. konfigurace výše)*

Tímto je konfigurace webového serveru kompletní.

## 3.2.6. Konfigurace databáze

Pro vytvoření databází je nutné se přepnout pod uživatele „postgres“ a zde spustit SQL konzoli „psql“. Přepnutí nejsnáze provedeme příkazem „su postgres“.

Poté otevřeme „psql“ a vložíme následující příkazy:

```
CREATE ROLE connectionpoint;
ALTER ROLE connectionpoint WITH NOSUPERUSER INHERIT NOCREATEROLE
NOCREATEDB LOGIN PASSWORD 'md52d5ce6268260d71d865927a940a10961';

CREATE DATABASE connectionpoint WITH TEMPLATE = template0 OWNER =
connectionpoint ENCODING = 'UTF8';

\connect connectionpoint

SET client_encoding = 'UTF8';
SET standard_conforming_strings = off;
SET check_function_bodies = false;
SET client_min_messages = warning;
SET escape_string_warning = off;

SET search_path = public, pg_catalog;

SET default_tablespace = '';

SET default_with_oids = true;

CREATE TABLE login (
    "loginId" integer NOT NULL,
    "userId" integer DEFAULT 0 NOT NULL,
    login text NOT NULL,
    password text NOT NULL,
    rights smallint DEFAULT 0 NOT NULL,
    locked smallint DEFAULT 0 NOT NULL,
    internet smallint DEFAULT 0 NOT NULL,
    "lastGranted" timestamp without time zone,
    "grantedIp" inet,
    "lastDenied" timestamp without time zone,
    "deniedIp" inet,
    "changeId" integer DEFAULT 0 NOT NULL,
    "changeDate" timestamp without time zone,
    "insertId" integer DEFAULT 0 NOT NULL,
    "insertDate" timestamp without time zone NOT NULL
);
```

*ALTER TABLE public.login OWNER TO connectionpoint;*

*CREATE SEQUENCE "login_loginId_seq"*
*INCREMENT BY 1*
*NO MAXVALUE*
*NO MINVALUE*
*CACHE 1;*

*ALTER TABLE public."login_loginId_seq" OWNER TO connectionpoint;*

*ALTER SEQUENCE "login_loginId_seq" OWNED BY login."loginId";*

*SELECT pg_catalog.setval('"login_loginId_seq"', 2, true);*

*ALTER TABLE login ALTER COLUMN "loginId" SET DEFAULT nextval('"login_loginId_seq"'::regclass);*

*INSERT INTO login VALUES (1, 0, 'admin', '7af8b85d8f4e182ecd86b836b1dae67077bc206f', 2, 0, 1, '2011-05-13 11:05:08', '192.168.77.2', '2011-05-12 19:05:43', '192.168.77.2', 0, NULL, 0, '2011-05-11 00:00:00');*
*INSERT INTO login VALUES (2, 0, 'test', 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3', 1, 0, 1, '2011-05-13 13:05:28', '192.168.77.80', NULL, NULL, 2, '2011-05-13 11:05:27', 2, '2011-05-13 11:05:18');*

Tím jsme vytvořili základní strukturu a uživatele. Je samozřejmě možné SQL příkazy volitelně upravil. Hesla loginů jsou uložena pomocí SHA1 otisků.

Pokud chceme, aby databáze ověřovala přístupy pomocí hesel, provedeme ještě jednu změnu.

*Nastavit v **/etc/postgresql/8.3/main/pg_hba.conf***

*# "local" is for Unix domain socket connections only*
***local all all md5***

## 3.2.7. Implementace webové aplikace

Webová aplikace jako taková je naprogramována v jazyce PHP a využívá pro směrování paketů a blokování uživatelů funkcí „iptables“. Proto je nutné, aby měla přístup k jejich ovládání, což není standardní. Zároveň vyžaduje přístup i k funkcím příkazu „arp“ kvůli ověření MAC adresy klienta. Tuto maličkost vyřešíme pomocí konfigurace funkce „sudo“, která umožní pro vybrané aplikace a uživatele pracovat v režimu správce. Implementace je jednoduchá pomocí konfigurace.

*Nastavit v **/etc/sudoers***

*# Apache access to iptables and ARP*
***www-data ALL=NOPASSWD: /sbin/iptables, /usr/sbin/arp***

Dále se musíme postarat o to, aby „iptables“ načetly výchozí konfiguraci po startu systému a aby se v jádru OS povolilo směrování paketů. To provedeme za pomocí skriptu a konfiguračního souboru. Nastavení „iptables“ je provedeno tak, že se ve výchozím stavu všechna TCP spojení přesměrovávají na port, kde poslouchá webový server a ten je tak může odchytit.

Veškerý provoz skrze server je aktivně zablokován. Parametr „—reject-with tcp-reset“ je použit proto, aby se případné špatně otevřené spojení rychle zavřelo a navázalo se nové na webový server.

*Soubor **/etc/firewall.conf***

**nat*
*:PREROUTING ACCEPT [38:2706]*
*:POSTROUTING ACCEPT [44:3402]*
*:OUTPUT ACCEPT [201:14360]*
***-A PREROUTING -i ! eth0 -p tcp -j DNAT --to-destination 192.168.77.1:80***
***-A POSTROUTING -o eth0 -j MASQUERADE***
*COMMIT*

**mangle*
*:PREROUTING ACCEPT [1876:481607]*
*:INPUT ACCEPT [1166:260677]*
*:FORWARD ACCEPT [710:220930]*
*:OUTPUT ACCEPT [1137:284396]*
*:POSTROUTING ACCEPT [1551:488142]*

*COMMIT*

**filter*
*:INPUT ACCEPT [1166:260677]*
***:FORWARD DROP [95:5700]***
*:OUTPUT ACCEPT [1137:284396]*
***-A FORWARD -j REJECT -p TCP --reject-with tcp-reset***
*-A FORWARD -j REJECT --reject-with icmp-net-prohibited*
*COMMIT*

*Soubor* ***/etc/network/if-up.d/iptables***

***#!/bin/sh***
***echo 1 > /proc/sys/net/ipv4/ip_forward***
***iptables-restore < /etc/firewall.conf***

Abychom měli ošetřené, že v „iptables“ nebudou zůstávat nepotřebné a nechtěné záznamy, je dobré jednou za čas provést vyčištění. Systém se postará, pokud dojde k ověření „Session“, o velmi rychlé znovu-zpřístupnění webu. Nejsnazším řešením je znovu načtení konfigurace, například jednou za hodinu. Je dobré provézt během toho i restart tabulky spojení, aby došlo k rychlejšímu nasměrování na obnovovací stránku.

*Soubor* ***/etc/cron.hourly/iptables***

***#!/bin/sh***
***echo 1 > /proc/sys/net/ipv4/ip_forward***
***iptables-restore < /etc/firewall.conf***
***conntrack -F***

Dále si stroj pojmenujeme, abychom nemuseli využívat v prohlížeči IP adresu. DNSmasq záznamy z tohoto souboru načítá také.

*Nastavit v* ***/etc/hosts***

***192.168.77.1 connection-point.web.***

Vytvoříme složku *„/var/www-sessions*“ a dáme jí plná oprávnění pro všechny.

Do složky *„/var/www*“ umístíme kód webové aplikace, který nalezneme na přiloženém CD (soubor: „*www.rar“*)

Do složek *„/var/www/web-desktop“* a *„/var/www/web-mobile“* umístíme prezentace společnosti pro mobilní zařízení a PC.

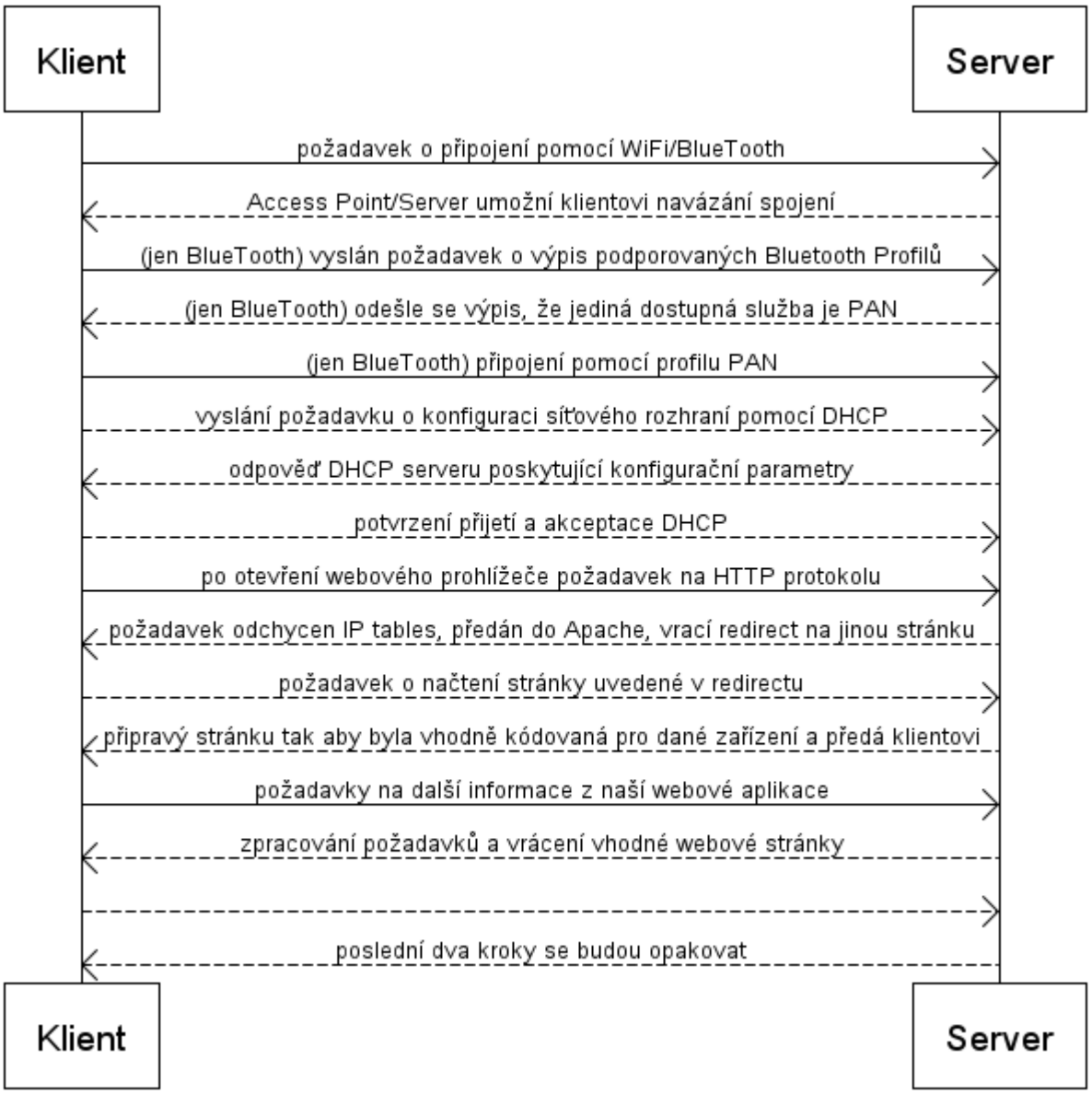
Nastavení by mělo být takto kompletní, proto provedeme restart systému a po připojení by se nám měl načíst informační web.

# 4. Popis funkce systému

Poté, co se vše podaří úspěšně provázat do jednoho funkčního systému, by mělo samotné připojení probíhat následujícím způsobem. Klient si na svém zařízení nechá vyhledat bezdrátové WiFi sítě, kde nalezne i tu, kterou jsme vytvořili v rámci našeho projektu. Následně se klient připojí, pomocí DHCP se v zařízení provedou všechna potřebná síťová nastavení a zařízení se bude tvářit jako připojené k internetu. Jakmile si klient otevře na zařízení internetový prohlížeč a vyšle požadavek na webovou stránku, bude tento požadavek odchycen na našem serveru a namísto stránky žádané, dojde k přesměrování na naši webovou aplikaci, která poskytne požadované informace automaticky ve správném formátu vhodném pro zařízení.

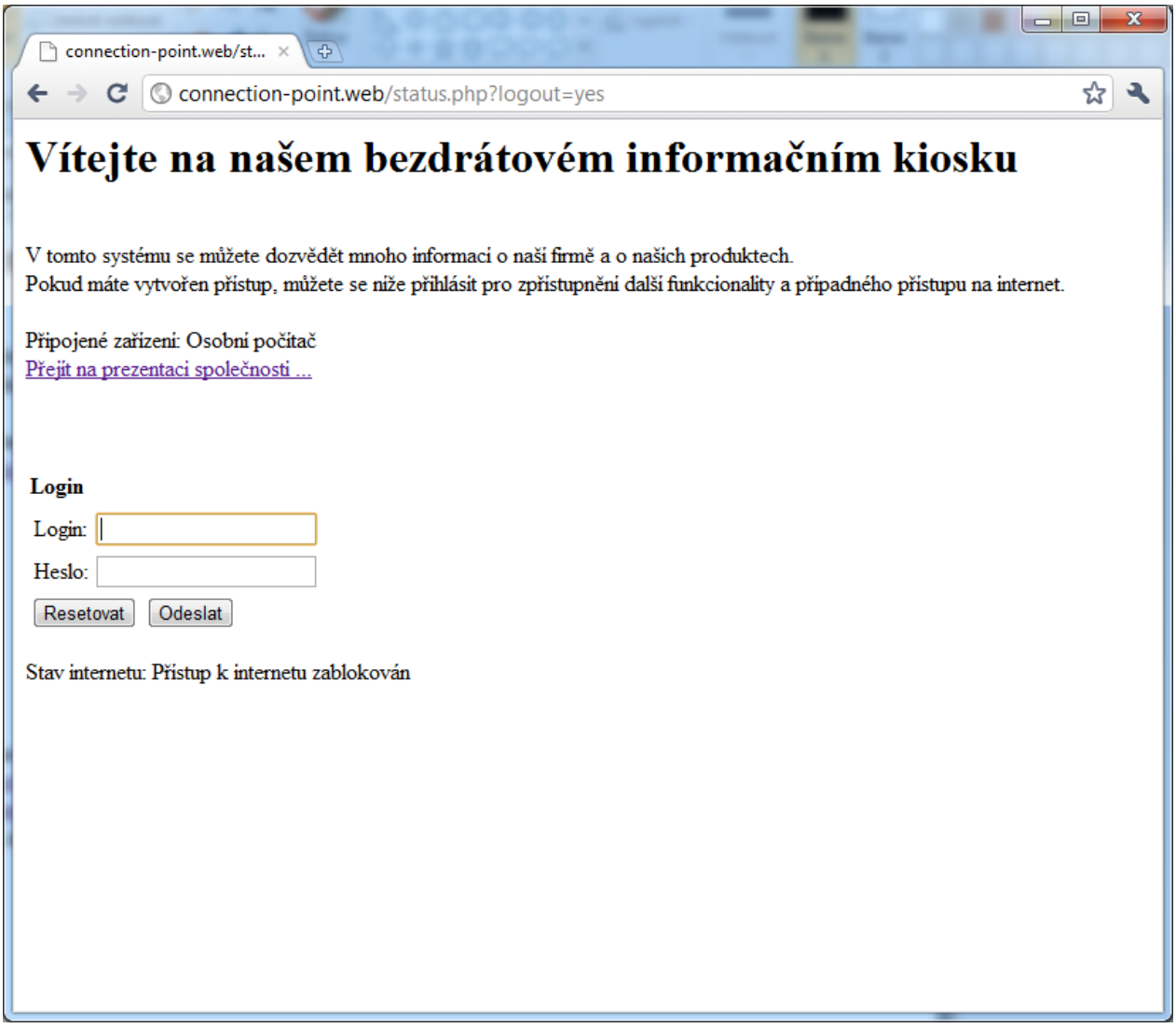
Pokud zařízení nebude podporovat WiFi, bude možné se připojit i pomocí BlueTooth. To proběhne tak, že při vyhledání okolních zařízení bude nalezen i náš server, který umožní navázání spojení a následné vybrání služby NAT (Network Access Point). Z důvodu nutnosti spárování, je nutné zadat PIN, který je přednastaven na hodnotu „1234“. Potom, co se služba připojí protokolem PAN, přidělí DHCP síťové nastavení a další postup je již stejný jako v případě WiFi.

Po technické stránce proběhne navázání komunikace následovně:



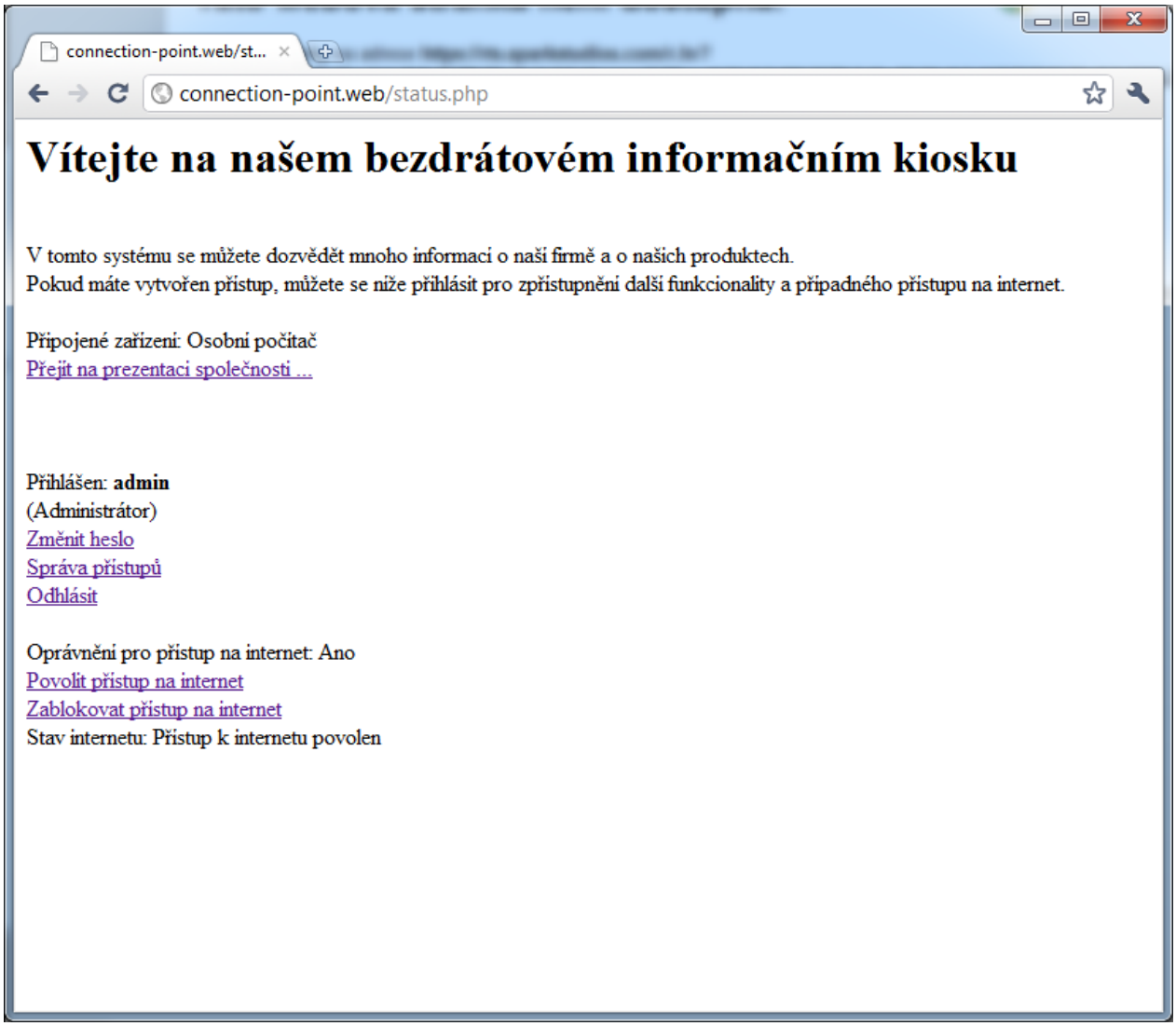
## 4.1. Představení webového rozhraní

Webová stránka portálu, která se nám načte po úspěšném navázání spojení a následném pokusu o přístup na internet vypadá následovně (Obrázek č.3).



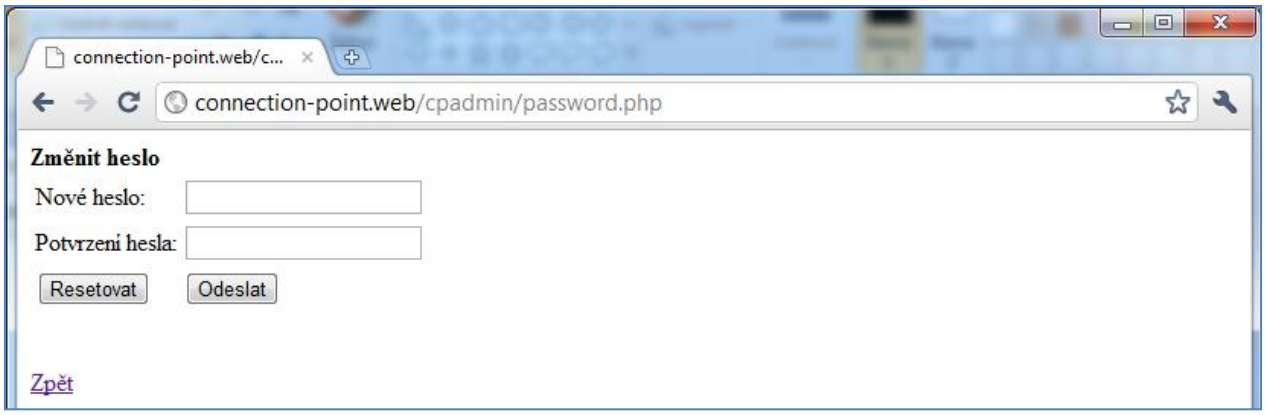

**Obrázek č.3**

Pokud má uživatel zřízen přístupový účet, může se do systému přihlásit a spravovat tak účty, případně si umožnit přístup na internet (Obrázek č.4).



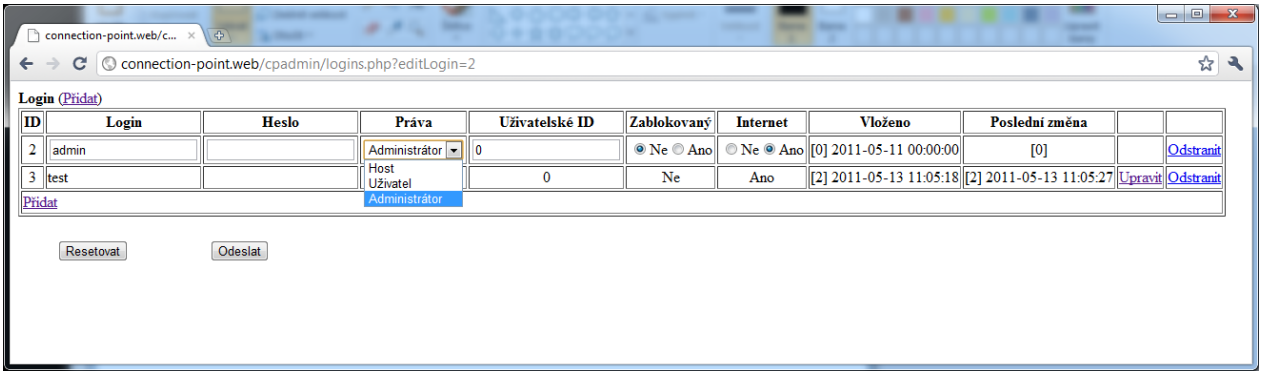

**Obrázek č.4**

Uživatelé si mohou po přihlášení měnit heslo. (Toto oprávnění nemají jen uživatelé s právy „host") (Obrázek č.5)



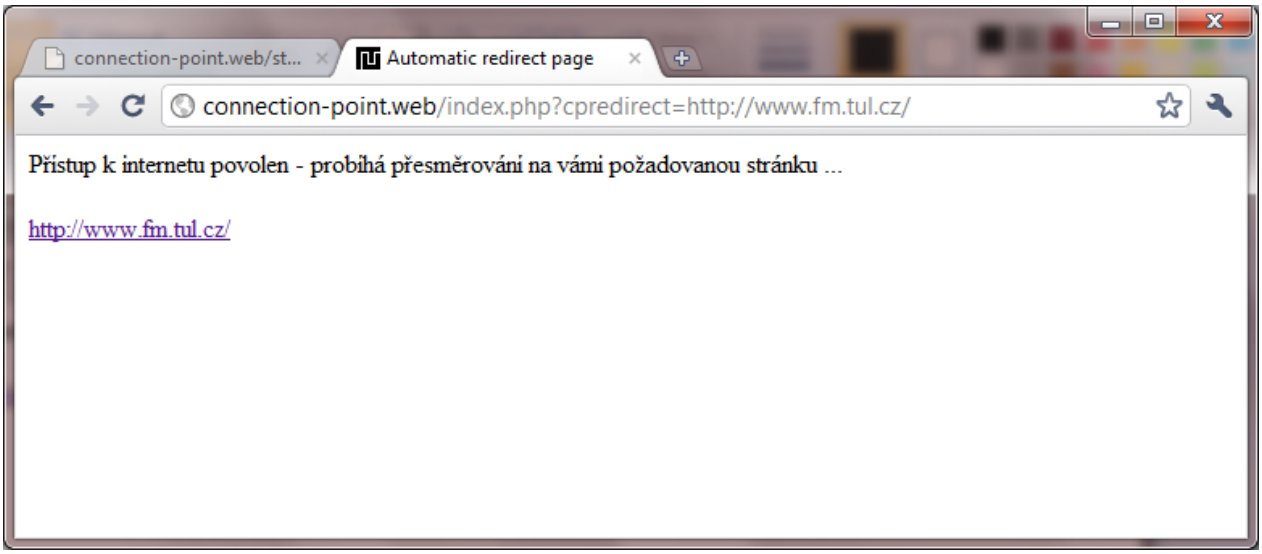

**Obrázek č.5**

Uživatelé s oprávněním „administrátor" mohou upravovat a přidávat i další uživatele. (Obrázek č.6)




**Obrázek č.6**

Pokud dojde k čištění „iptables“ a uživatel je oprávněn přistupovat na internet, je provedeno znovu ověření pomocí „Sessions“ a obnovení pravidel. Obnovovací stránka vypadá následovně. (Obrázek č.7, Obrázek č.8)




**Obrázek 7**


**Obrázek 8**

# 5. Závěr

Cílem práce bylo navrhnout systém bezdrátových informačních kiosků, jako alternativu pro kiosky statické. To se podařilo zrealizovat v plné míře, i přes menší potíže s nastavením BlueTooth zařízení.

Relativně hodně času zabralo také odlazení přesměrování při znovu ověření dle „Session“, ale nakonec vše pracuje jak má a uživatele tento zásah v podstatě nijak neomezí. Problém by mohl nastat jen v případě stránek provozovaných přes protokol HTTPS, kde může dělat problém certifikát. Jelikož ale systém nemá sloužit jako dlouhodobý prostředek pro připojení a obnovení nastavení „iptables“ (pročištění) je prováděno jen jednou za hodinu, neviděl bych to jako velký problém.

Systém bezdrátových informačních kiosků se povedlo zcela zrealizovat za pomocí OpenSource součástí, náklady na software jsou tedy minimální, navíc je možné využít přiložený virtuální obraz pro VirtualBox a odpadají tak náklady i na čas strávený instalací.

Dle testování se systém tváří velice dobře, spojení jsou okamžitě po přístupu prohlížeče odchycena. Celkově bylo tedy zadání úspěšně splněno a systém je možné využívat.

# Použité zdroje


[1] [Vít Svatopluk]
Linuxové distribuce letmým začátečnickým pohledem
[online] URL: http://www.root.cz/clanky/linuxove-distribuce-letmym-zacatecnickym-pohledem/

[2] [The Apache Software Foundation]
Apache HTTP Server Version 2.2 Documentation
[online] URL: http://httpd.apache.org/docs/2.2/

[3] [Wikimedia Foundation]
IEEE 802.11
[online] URL: http://cs.wikipedia.org/wiki/IEEE_802.11

[4] [Wikimedia Foundation]
IEEE 802.16
[online] URL: http://cs.wikipedia.org/wiki/IEEE_802.16

[5] [Wikimedia Foundation]
IEEE 802.20
[online] URL: http://cs.wikipedia.org/wiki/IEEE_802.20

[6] [Wikimedia Foundation]
OFDM
[online] URL: http://cs.wikipedia.org/wiki/OFDM

[7] [Wikimedia Foundation]
WiMAX
[online] URL: http://cs.wikipedia.org/wiki/WiMAX

[8] [Wikimedia Foundation]
Wi-Fi Protected Access
[online] URL: http://cs.wikipedia.org/wiki/WPA

[9] [Wikimedia Foundation]
IrDA
[online] URL: http://cs.wikipedia.org/wiki/IrDA

[10] [Wikimedia Foundation]
Bluetooth profile
[online] URL: http://en.wikipedia.org/wiki/Bluetooth_profile

[11] [Wikimedia Foundation]
Transport Layer Security
[online] URL: http://cs.wikipedia.org/wiki/TLS

[12] [Bouresh Zdenek]
Bluetooth PAND (Personal Area Network) Howto For Debian Etch
[online] URL: http://www.howtoforge.com/bluetooth_pand_debian_etch

[13] [NetBeans Community, Oracle Corporation]
NetBeans IDE 7.0 Download
[online] URL: http://netbeans.org/downloads/index.html

[14] [Kelley Simon]
dnsmasq - A lightweight DHCP and caching DNS server
[online] URL: http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq-man.html

[15] [PostgreSQL Global Development Group]
PostgreSQL - Manual
[online] URL: http://www.postgresql.org/docs/8.3/static/

[16] [PHP Group]
PHP5 - Manual
[online] URL: http://php.net/manual/en/

[17] [PHP Group]
PHP5 - Manual - Sessions
[online] URL: http://php.net/manual/en/book.session.php

[18] [PHP Group]
PHP5 - Cookies - Manual
[online] URL: http://php.net/manual/en/function.setcookie.php

[19] [PHP Group]
PHP5 - Headers - Manual
[online] URL: http://php.net/manual/en/function.header.php

[20] [PHP Group]
PHP5 - get_browser - Manual
[online] URL: http://php.net/manual/en/function.get-browser.php

[21] [Tackmann Lars]
Getting IPTables to survive a reboot
[online] URL: http://www.debian-administration.org/articles/445

[22] [Herve Eychenne and others]
iptables(8) - Linux man page
[online] URL: http://linux.die.net/man/8/iptables

[23] [Grygárek Petr, FEI, VŠB-TU Ostrava]
Konfigurace překladu adres (NAT) s pomocí Linux IPTables
[online] URL: http://www.cs.vsb.cz/grygarek/PS/projekt0405/NAT-priklad.pdf

[24] [Gary J. Keith and the Browser Capabilities Project]
Browser Capabilities Project
[online] URL: http://browsers.garykeith.com/downloads.asp

[25] [Vivek Kapoor]
Execute system commands via PHP
[online] URL: http://exain.wordpress.com/2007/11/24/execute-system-commands-via-php/

[26] [Vitek Gite]
/etc/network/interfaces Ubuntu Linux networking example
[online] URL: http://www.cyberciti.biz/faq/setting-up-an-network-interfaces-file/

[27] [Ring of Saturn Internetworking]
IPTables Example Config
[online] URL: http://networking.ringofsaturn.com/Unix/iptables.php

[28] [Debian GNU/Linux]
Debian - List of sections in "lenny"
[online] URL: http://packages.debian.org/lenny/

[29] [Janovský Dušan]
HTML příručka
[online] URL: http://www.jakpsatweb.cz/html/

[30] [Wiles Frank]
Quick-Tip: Linux NAT in Four Steps using iptables
[online] URL: http://www.revsys.com/writings/quicktips/nat.html

[31] [Burt Adam]
Sample wap page
[online] URL: http://homepage.mac.com/a.burt/3g/mlearning/how_to/Sample.htm

[32] [Berry Wayne]
Sharing Cookies Across Domains
[online] URL: http://www.15seconds.com/issue/971108.htm

[33] [Linux Report]
iptables Tutorial
[online] URL: http://www.linuxreport.org/content/view/26/23/

[34] [Billauer Eli]
IP Masquerading using iptables
[online] URL: http://www.billauer.co.il/ipmasq-html.html

[35] [The CentOS team]
IPTables
[online] URL: http://wiki.centos.org/HowTos/Network/IPTables

[36] [Debian Admin]
Providing root privileges for users Using SUDO
[online] URL: http://www.debianadmin.com/providing-root-privileges-for-users-using-sudo.html

[37] [Cornevilli Fabio, Cinque Marcello]
Bluetooth NAP How To
[online] URL: http://www.mobilab.unina.it/BlueNAPHOWTO.htm

[38] [Schmidt Michael, University of Siegen, Germany]
HowTo set up common PAN scenarios with BlueZ's integrated PAN support
[online] URL: http://bluez.sourceforge.net/contrib/HOWTO-PAN

[39] [Gentoo Linux Wiki]
Bluetooth Network Aggregation Point
[online]
URL: http://en.gentoo-wiki.com/wiki/Bluetooth_Network_Aggregation_Point

[40] [Csaba Botoš]
Seriál Vše o iptables
[online] URL: http://www.root.cz/serialy/vse-o-iptables/